APPENDIX I

Exemplary Semantic Specifications of the LAPI functions LAPI_Purge_totask and LAPI_Resume_totask.

LAPI_Purge_totask

Purpose: To allow a task to cancel/purge messages to a given destination

C Syntax

```
int
LAPI_Purge_totask(hndl, dest)
lapi_handle_t  hndl;
task_t         dest;
```

FORTRAN Syntax

```
include ``lapif.h"
int
LAPI_PURGE_TOTASK(HNDL, DEST, IERROR)}
INTEGER      hndl;
INTEGER      dest;
INTEGER      ierror;
```

Parameters

hndl	IN	handle to the LAPI context.
dest	IN	the destination instance id to which pending messages need to be canceled.

Description

This function cancels messages and resets the state corresponding to messages in flight or submitted to be sent to a particular instance. Note that this is an entirely local operation. For correct behavior a similar invocation is expected on the destination (if it exists). This function cleans up all of the state information associated with pending messages to the indicated instance. It is assumed that before the indicated instance starts communicating with this instance again, it also does a "cancel" to this instance (or that it was terminated and initialized again). It also wakes up any or all threads that are in {\bf LAPI_Nopoll_wait()} depending on how the arguments are passed to the {\bf LAPI_Nopoll_wait()} function. The behavior of {bf LAPI_Purgue_totask()} is undefined if {/bf LAPI} Global functions are utilized. If the user wants to avoid any possible trickle traffic that exists in the network, the user should also call

LAPI_Send(hndl,EPOCH_NUM,<new epoch_num>) function to ensure LAPI does not receive any old message that may exist from the Purged task. This resets the global state; thus, all nodes that are still "up" should make the same call, otherwise communication packets are very likely to be dropped. Also the node coming up should utilize the new *epoch_num* as part of initializing the *lapi_info_t*

structure for the **LAPI_Init** (lapi_handle_t, lapi_info_t) call. If **LAPI_Purge_totask**() is used, then once the purged task comes up the user should call **LAPI_Resume_totask**() to be able to send messages to the reinitialized task. Also, if LAPI_Purge_totask() is called and subsequent **LAPI** communication calls are made to the task that were purged, those calls are returned with the following error code: LAPI_ERR_PURGED_TASK.

Return Values

LAPI_SUCCESS - if the state was successfully reset.
LAPI_ERR_BAD_PARAMETER - invalid parameter passed in.

Related Information

LAPI_Resume_totask, LAPI_Nopoll_wait, LAPI_Init

LAPI_Resume_totask

Purpose: To re-enable messages to be sent to the purged task.

C Syntax

```
#include <lapi.h>
int
LAPI_Resume_totask(hndl, dest)
lapi_handle_t     hndl;
task_t            dest;
```

FORTRAN Syntax

```
include ``lapif.h"
int
LAPI_RESUME_TOTASK(HNDL, DEST, IERROR)
INTEGER      hndl;
INTEGER      dest;
INTEGER      ierror;
```

Parameters

hndl	IN	handle to the LAPI context.
dest	IN	the destination instance id with which it wants to communicate again

Description

This function re-enables messages to be sent to the purged task. If some task goes down and this task calls LAPI_Purged_totask() it calls {/bf LAPI_Resume_totask()} if it wants to communicate with that task again. The task that went down must come back up for the communication to occur.

Return Values

LAPI_SUCCESS - if the state was successfully reset.
LAPI_ERR_BAD_PARAMETER - invalid parameter passed in.

Related Information

LAPI_Purge_totask, LAPI_Nopoll_wait, LAPI_Init

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

09505308.071301

Application deficiencies found during scanning:

☐ Page(s)__________ of Express mail __________ were not present for scanning.
(Document title)

☐ Page(s) 2 of Specifications __________ were not present for scanning.
(Document title)

☐ ***Scanned copy is best available.***